Quantiles

---

29 marks

1. Suppose we have a continuous random variable X with distribution function $F_X(x) = Pr(X \le x)$ and quantile function $Q_X(p) = F_X^{-1}(p)$. That is $p = F_X(x) = Pr(X \le x)$ and $p = Pr(X \le Q_X(p)) = F_X(Q_X(p)) = F_X(F_X^{-1}(p)) = p$.

   a. *(4 marks)* Suppose $Y = aX + b$ for some constants $a > 0$ and b. **Prove** that a plot of the parametric curve $(Q_X(p), Q_Y(p))$ for $p \in (0, 1)$ must follow a straight line.

   Give the equation of that line.

$$p = Pr(Y \le Q_Y(p)) = Pr(aX + b \le Q_Y(p)) = F_X(\frac{Q_Y(p) - b}{a})P = F_X(Q_X(p))$$

   Therefore

$$\frac{Q_Y(p) - b}{a} = Q_X(p)Q_Y(p) = aQ_X(p) + b$$

   b. *(3 marks)* When $F_X(x)$ and $Q_X(p)$ are the cumulative distribution and quantile functions of the continuous random variable X, show that if $U \sim U(0, 1)$, then

$$Pr(Q_X(U) \le x) = F_X(x).$$

   We have $U \sim U(0, 1)$ so that $Pr(U < a) = a$ for $a \in [0, 1]$ therefore

$$Pr(Q_X(U) \le x) = Pr(F_X(Q_X(U)) \le F_X(x)) = Pr(U \le F_X(x)) = F_X(x)$$

   c. The above result implies that we could generate n independently and identically distributed (i.i.d.) random realizations X from $F_X(x)$ by generating n i.i.d. random realizations U from $U(0, 1)$ and defining $X = Q_X(U)$.

   In R the function `runif()` will generate uniform pseudo-random numbers.

   (Similarly, `dunif()`, `punif()`, and `qunif()` will return the density, the distribution, and the quantile functions, respectively, for a uniform random variable. See `help("runif")` for details.

      i. *(1 mark)* Write an R function

```
r_unifgenFx <- function(n, qfunction = qnorm) {
                    # Insert your code here
                    random <- runif(n)
                    qfunction(random)
                     }
```

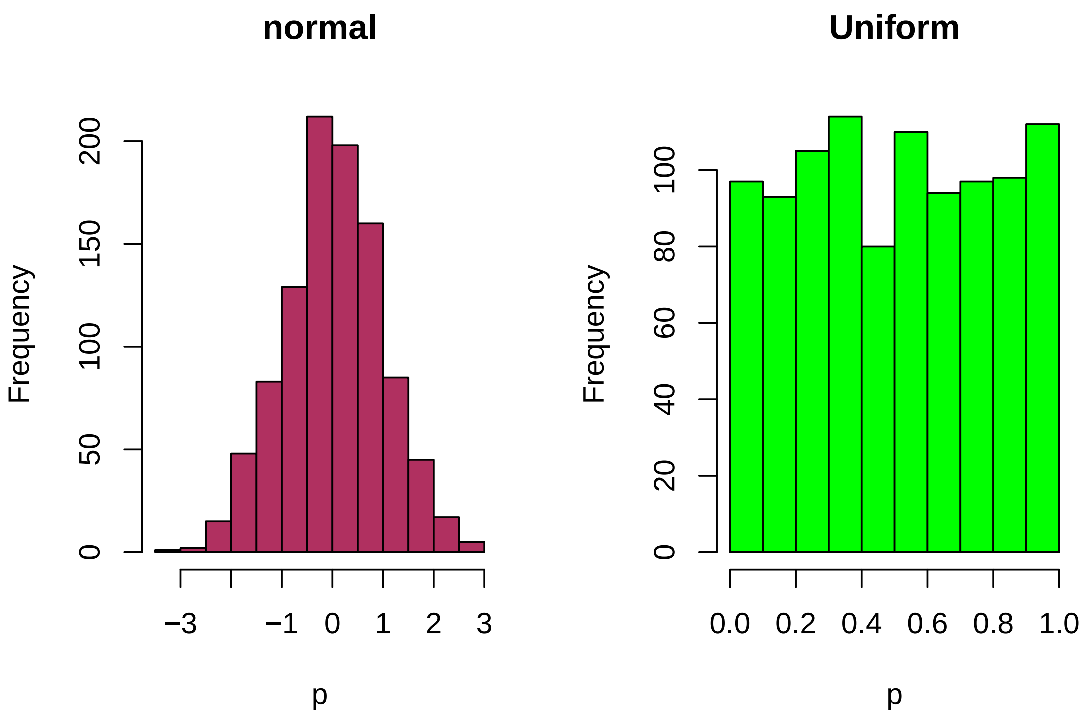
      which will generate and return `n` pseudo random observations from the distrubution whose quantile function is the value of the argument `qfunction`. Show your code.

      ii. *(2 marks)* Execute the following code snippets to illustrate your code

```r
# make sure we all get the same result
set.seed(1234567)
# save the currentgraphical parameters and set `mfrow`
oldPar <- par(mfrow = c(1,2))

hist(r_unifgenFx(1000),main = "normal", xlab="p", col = 'maroon', border="black")  # Standa

hist(r_unifgenFx(1000, qfunction = qunif), main = 'Uniform', xlab = 'p', col = 'green',bord
```




```r
par(oldPar) # Return to original graphical parameters
```

iii. *(2 marks)* Generate a sample of 1000 pseudo-random observations from a Student-t distribution on 3 degrees of freedom generated using `r_unifgenFx()` (unchanged) and the quantile function of the Student-t. Plot a histogram (appropriately labelled) of the results.

````
```r
set.seed(1234567)
oldPar <- par(mfrow = c(1,2))
hist(qt(r_unifgenFx(1000, qfunction = qunif), df=3), main = 'Student-t distribution', xlab =
par(oldPar) # Return to original graphical parameters
```

![](quantiles_files/figure-latex/hist_t-1.pdf)<!-- -->
````

d. Consider the `quantile()` function in R .

i. *(2 marks)* Explain the values returned by `quantile(mtcars$qsec))`. That is, what does `quantile()` do? A quantile, or percentile, tells you how much of your data lies below a certain value.

```r
quantile(mtcars$qsec)
```

```
##      0%     25%     50%     75%    100%
## 14.5000 16.8925 17.7100 18.9000 22.9000
```

For example, if we call quantile(mtcars$qsec), it returns that the first 25% values are in (14.5000, 16.8925), the second 25% of the values are in (16.8925, 17.7100). The third 25% values are in (17.7100, 18.9000) and the last 25% values are in (18.9000,22.9000)

ii. *(2 marks)* Show how `quantile()` could be used to generate 1000 observations from the estimated distribution of `mtcars$qsec`.

```r
estimated_qsec <- quantile(mtcars$qsec, probs = runif(1000, min = 0, max =1))
```

````
iii. *(2 marks)* Would this work for `mtcars$cyl`? Why? Or, why not?
  No. The reason is the values if mtcars$cyl can be 4, 6, and 8 only. We cannot divide them into 4 
iv. *(4 marks)* Draw side by side (nicely labelled) histograms of `mtcars$qsec` and a sample of 100

  ```r
  set.seed(1234567)
    oldPar <- par(mfrow = c(1,2))
    hist(mtcars$qsec,main = 'quarter-mile seconds', xlab = 'Seconds', col = 'yellow')
    hist(estimated_qsec, main = 'Estimated seconds', xlab = 'Seconds', col = 'maroon')
  ```

  ![](quantiles_files/figure-latex/unnamed-chunk-3-1.pdf)<!-- -->

  ```r
    par(oldPar)
  ```

  I think the estimation is very close to true.

v. *(3 marks)* Draw a (nicely labelled) `qqplot()` comparing the above two sets of observations.  W

  ```r
    qqplot(mtcars$qsec, estimated_qsec, main = 'qqplot, quarter-mile seconds and estimated distribu
  ```

  ![](quantiles_files/figure-latex/unnamed-chunk-4-1.pdf)<!-- -->
  The estimation is percise because the y-intercept of the plot is close to 0 and the slope is clos

vi. *(4 marks)* Suppose interest lay in producing a bootstrap distribution for some estimator $\big
  We can further use the quantile function for each 25% of the data. By making 4x4=16 quantiles, th
  Bootstraping is more complex and the result may depend on the representative sample.
````